CLAIMS

What is claimed is:

1. A method comprising:

separating a random number generator into a first stage and second stage;

replicating the circuitry of said second stage into M parallel modules; and

generating M random number outputs using a single first stage circuitry and M parallel second stage modules.

2. A method according to claim 1 wherein each of said M modules have a first input which is the output result of said single first stage circuitry and second input, which is the unique to the module.

3. A method according to claim 2 wherein said first stage circuitry has at least two first stage inputs, a first first stage input representing the integer coordinate components of the location of interest and a second first stage input representing the hardware cycle.

4. A method according to claim 3 wherein generating includes:

storing said first stage inputs in a first register;

selecting a first set of bits from said first register;

storing said first set of bits in a second register; and

selecting a second set of bits from said second register, said second set of bits representing said result of said single first stage circuitry.

5. A method according to claim 4 wherein for each parallel second stage module, generating further includes:

storing said result of said single stage circuitry in a second stage register; and

selecting a second set of bits from said second stage register, said second set of bits representing one of said M random number outputs.

6. A method according to claim 5 wherein said each random number output can be one of a phase 1 output and a phase 2 output.

7. A method according to claim 6 wherein said phase 2 output is a full and exact representation of the desired random number output.

8. A method according to claim 7 wherein said phase 1 output is an approximation of said phase 2 output.

9. A method according to claim 4 wherein selecting said first set of bits includes:

utilizing a group of XORs, each XOR having its inputs pre-wired to various locations of said first register, to select among the bits of said inputs.

10. A method according to claim 4 wherein selecting said second set of bits includes:

utilizing a group of XORs, each XOR having its inputs pre-wired to various locations of said second register, to select among the bits stored therein.

11. A method according to claim 5 wherein said selecting in each parallel second stage module includes:

utilizing a group of XORs, each XOR having its inputs wired to a number of bits representing the module number and to selected locations of said second stage register.

12. A method according to claim 11 wherein the wired number of bits representing the module number varies from XOR to XOR.

13. A method according to claim 11 wherein each parallel second stage module wires its XORs to its registers in an identical manner with all other parallel second stage modules.

14. A method of generating a noise function output at a first location comprising:

generating M random number outputs, each random number output representing a pulse, using a single first stage and M parallel second stage circuitries;

determining the relevance of each random number output to said noise function output; and

accumulating each of an effective weighting computed for each relevant random number output.

15. A method according to claim 14 wherein generating includes:

storing said first stage inputs in a first register;

selecting a first set of bits from said first register;

storing said first set of bits in a second register; and

selecting a second set of bits from said second register, said second set of bits representing said result of said single first stage circuitry.

16. A method according to claim 15 wherein generating, for each parallel second stage module, further includes:

storing said result of said single stage circuitry in a second stage register; and

selecting a second set of bits from said second stage register, said second set of bits representing one of said M random number outputs.

17. A method according to claim 16 wherein said each random number output can be one of a phase 1 output and a phase 2 output.

18. A method according to claim 17 wherein said phase 2 output is a full and exact representation of the desired random number output.

19. A method according to claim 18 wherein said phase 1 output is an approximation of said phase 2 output.

20. A method according to claim 15 wherein selecting said first set of bits includes:

utilizing a group of XORs, each XOR having its inputs pre-wired to various locations of said first register, to select among the bits of said inputs.

21. A method according to claim 15 wherein selecting said second set of bits includes:

utilizing a group of XORs, each XOR having its inputs pre-wired to various locations of said second register, to select among the bits stored therein.

22. A method according to claim 16 wherein said selecting in each parallel second stage module includes:

utilizing a group of XORs, each XOR having its inputs wired to a number of bits representing the module number and to selected locations of said second stage register.

23. A method according to claim 19 wherein determining the relevance includes:

testing phase 1 outputs to see if the pulses they represent fall approximately within an area of interest about said first location; and

updating a queue with a positive indication for each pulse that falls within said area.

24. A method according to claim 23 wherein said area of interest is determined by a circle swept by a radius about said first location.

25. A method according to claim 24 wherein testing includes:

determining the distance between the location of a pulse as approximated by their phase 1 output and said first location;

subtracting an allowable difference from said determined distance; and

comparing the result of said subtraction with said radius.

26. A method according to claim 25 wherein if said result is not greater than said radius, then said pulse is determined to fall approximately within said area of interest.

27. A method according to claim 19 wherein accumulating includes:

fetching the full phase 2 output for each pulse determined to be relevant to said noise function output;

determining what effect each said relevant pulse would have to said noise function;

modifying by said effect the weight of said pulse given by said phase 2 output, said modification yielding an effective weight; and

summing together effective weights for all pulses deemed to be relevant to said noise function output, said sum representing the amplitude of the noise function at said first location.

28. A method according to claim 27 wherein determining what effect includes:

computing a sum of squared differences between said pulse as represented by its phase 2 output and said location;

correlating said sum of squared differences with a corresponding effect factor; and

obtaining said effect factor from a filter table.

29. A method according to claim 28 wherein modifying includes:

multiplying said effect factor by said weight.

30. An apparatus configured to generate M random number outputs on a given cycle j in a parallel fashion, comprising:

a first stage circuitry configured to accept a series of inputs; and

M second stage modules, the output of said first stage wired as an input each of said M modules, the output of each module one of said M random number outputs.

31. An apparatus according to claim 30 wherein said series of inputs includes the cycle number j, the number of dimensions desired and integer coordinate components for each dimension desired.

32. An apparatus according to claim 30 wherein each said second stage module also includes an input k representing the number of the module.

33. An apparatus according to claim 31 each said random number output contains a fractional coordinate component for each desired dimension as well as a weight of the corresponding pulse at the location given said fractional coordinate components and said integer coordinate components.

34. An apparatus according to claim 30 wherein said first stage circuitry comprises:

a first register having locations storing each bit of said series of inputs; and

a first bank of XORs, each input of each said XOR of said first bank coupled to an arbitrary one of said locations of said first register.

35. An apparatus according to claim 34 wherein said first stage circuitry further comprises:

a second register having locations storing each output of said first bank of XORs; and

a second bank of XORs, each input of said XORs of said second bank coupled to an arbitrary one of said locations of said second register, each output of said second bank of XORs representing each bit of said output of said second stage.

36. An apparatus according to claim 32 wherein each second stage module comprises:

a first register having locations configured to store each bit of said output of said first stage; and

a bank of XORs having one set of inputs from said first register and a second set of inputs from the bits of input k, the output of each XOR representing one bit of said random number output.

37. An apparatus according to claim 36 wherein said M second stage modules produce an approximation of the full said random number output.

38. An apparatus for generating a noise function output, said noise function output a combination of relevant pulses, comprising:

a random number generator configured to generate M random number outputs in a parallel fashion, each of said M

outputs an approximation of a full version of each of said random number outputs;

a chooser device coupled to said random number generator, said chooser determining if the pulse represented by each said random number output is relevant to the noise function;

a queue coupled to said chooser device, said queue storing which of said M random number outputs were determined relevant, said apparatus configured to generate the full version of the relevant random number outputs;

an effect generator coupled to said queue, said effect generator modifying the weight of each pulse as given by said full version according to the location of the pulse represented by said full version;

and an accumulation device coupled to said effect generator to accumulate said effect modified pulses weights, the result of said accumulation the amplitude of the noise function.